UNIT 17 DATA WAREHOUSING AND DATA MINING

Structure

17.1 INTRODUCTION

Over the past couple of decades huge investments have been made in computer systems by businesses of all types to gain competitive advantage. A large majority of these computer applications designed and developed during this period have concentrated on automating business processes, for example: order processing, inventory and itemized billing of telephone calls. The automated business systems, or the *operational systems* – as they are called, were not only good at doing what they were designed for, but also (possibly unintentionally) ended up collecting huge volumes of data. However, in the context of the twenty first century, competitive advantage comes less from mere automation (or even optimization) of day-to-day activities, and more and more from proactive strategic decisions based on analytical use of data. Unfortunately, the technological requirements of systems for supporting analytical applications, like *on-line analytical processing (OLAP)* and *data mining*, differ greatly from the requirements laid down when the operational systems were designed. We will discuss these differences in the next section and then go on to see how a new breed of systems, known as *data warehouses* have evolved, matured, and taken the world of business by storm in recent years.

17.2 OBJECTIVES

After reading this unit you should be able to:

- Identify the types of data in data warehouse;
- Describe the design decisions involved in data warehouse design;
- Explain the importance of building an Enterprise Data Warehouse;
- Depict the architecture of a data warehouse and its various components;
- Describe various aspects of Star and Snowflake schemas;
- Define and explain the concept of metadata in the data warehouse;
- Enumerate the importance of external data feed from the World Wide Web to the Data Warehouse;
- Describe the process of data analysis and on-line analytical processing;
- Develop a case for the business value of visualization of data; and
- Apply data mining tools for discovery of hidden relationships and patterns in business data.

17.3 OPERATIONAL AND INFORMATIONAL DATABASES

There are several reasons that have compelled information system designers to accept that even the latest technology cannot be optimized to support both operational and informational (or analytical) processing in a cost effective manner.

- The data required is different
- The technology required is different
- The user community is different
- The access pattern is different
- The processing characteristics are different
- The system loading pattern is different.

High level business objectives provide the framework for competitiveness and growth to an enterprise in the long run. To set realistically achievable objectives and monitor the company's performance against them, managers need information on *subjects* like customers (e.g. their needs and preferences), products (e.g. the company's offering vis-à-vis competitors), suppliers (e.g. their production lead times) and distributors (e.g. location). It is also beneficial to *aggregate* measures like quantities of products and sales values across various *dimensions* like geographical regions and time, and *compare* them with yard-sticks like budgets, forecasts, previous year's actuals and industry averages. The differences between informational systems satisfying needs of these types and traditional operational systems are summarized in the *Table 17.1*.

Table 17.1: Operational and Decision Support Data

PRIMITIVE DATA / OPERATIONAL DATA	DERIVED DATA / DSS DATA
• application oriented	• subject oriented
• detailed	• summarized , otherwise refined
• accurate, as of the moment of access	• represents values over time, snapshots
• serves the clerical community	• serves the managerial community
• can be updated	• is not updated
• run repetitively	• run heuristically
• requirements for processing understood a priori	• requirements for processing not understood a priori
• compatible with the SDLC	• completely different life cycle
• performance sensitive	• performance relaxed
• accesses a unit at a time	• accessed a set at a time
• transaction driven	• analysis driven
• control of update a major concern in terms of ownership	• control of update no issue
• high availability	• relaxed availability
• managed in its entirety	• managed by subsets
• nonredundancy	• redundancy is a fact of life
• static structure; variable contents	• flexible structure
• small amount of data used in a process	• large amount of data used in a process
• supports day-to-day operations	• supports managerial needs
• high probability of access	• low, modest probability of access

Information systems have evolved from the master-file / transaction file and sequential processing of the 1960s to DBMS based random access processing of the 1970s to high-performance transaction processing during the 80s to decision support systems built around the existing operational databases. The traditional IT departments when faced with the need to provide strategic information using data in operational databases find themselves inadequately equipped to respond.
Some of the difficulties faced are:

- IT receives too many ad hoc requests, resulting in a large overload. With limited resources, IT is unable to respond to the numerous requests in a timely fashion.
- Requests are not only too numerous, they also keep changing all the time. The users need more reports to expand and understand the earlier reports.
- The users find that they get into the spiral of asking for more and more supplementary reports, so they sometimes adapt by asking for every possible combination, which only increases the IT load even further.
- The users have to depend on IT to provide the information. They are not able to access the information themselves interactively.
- The information environment ideally suited for strategic decision-making has to be very flexible and conducive for analysis .IT has been unable to provide such an environment.

The solution to the problems lies in a new paradigm – the *data warehouse*. It is an *environment* that is specifically designed to hold data required to support complex analysis, and detection of patterns, trends and deviations.
The characteristics of such an environment can be summarized as follows:

- Provides an integrated and total view of the enterprise
- Make the enterprise's current and historical information easily available for decision-making.
- Makes decision-support transactions possible without hindering operational systems
- Renders the organization's information consistent.
- Presents a flexible and interactive source of strategic information.

17.4 THE DATA WAREHOUSE

We have seen in the previous section that we need a new breed of information delivery environment, called a data warehouse, to facilitate strategic decision-making. The concept of a data warehouse given by Bill Inmon, the father of data warehousing, is depicted in *Figure 17.1.*

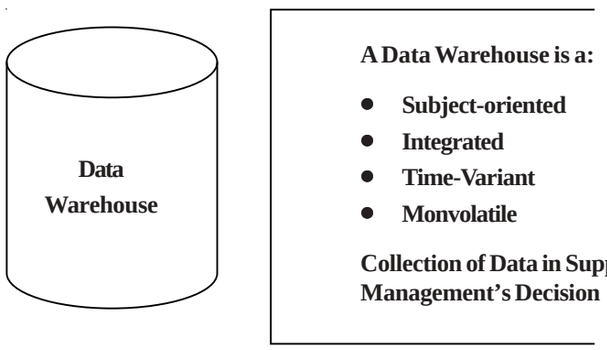


Fig. 17.1: What is a Data Warehouse?

The defining characteristics of a data warehouse are:

- **Subject-orientation**: Data warehouse data are arranged and optimized to provide answers to questions coming from diverse functional area within a company. Therefore, the data warehouse contains data organized and summarized by topic, such as sales, marketing, finance, distribution, and transportation. For each one of these topics the data warehouse contains specific subjects of interest - products, customers, departments, regions, promotions, and so on. Note that this form of data organization is quite different from the more functional or process-oriented organization of typical transaction systems.
- **Time-variancy:** We have already noted that the DSS data include a time element (see *Table 17.1*). In contrast to the operational data, which focus on current transactions, the warehouse data represent the flow of data through time. The data warehouse can even contain projected data generated through statistical and other models.
- **Non-volatility:** Once data enter the data warehouse they are never removed. Because the data in the data warehouse represent the company's entire history, the operational data representing the near-tern history, are always added to it. Because data are never deleted and new data are always added, the data warehouse is always growing. That is why the DSS DBMS must be able to support multi-gigabyte and even multi-terabyte database and multiprocessor hardware.
- **Integration:** The data warehouse is a centralized, consolidated database that integrates data derived from the entire organization. Thus the data warehouse consolidates data from multiple and diverse sources with diverse formats. Data integration implies a well-organized effort to define and standardize all data elements. This integration effort can be time-consuming but, once accomplished, it provides a unified view of the overall organizational situation. Data integration enhances decision-making and helps managers to better understand the company's operations. This understanding can be translated into recognition of strategic business opportunities.

Table 17.2 summarizes the differences between the data in a data warehouse and that in an operational database.

Table 17.2: Operational Data and Data Warehouse Data

CHARCTERISTIC	OPERATIONAL DATABASE DATA	DATA WAREHOUSE DATA
Integrated	Similar data can have different representations or meanings. For example, telephone numbers may be stored as 033-29-70701 or as 0332970701, and a given condition may be labeled as T/F or 0/1 or Y/N. A sales value may be shown in thousands or in millions.	Provide a unified view of all data elements with a common definition and representation for all business units.
Subject-oriented	Data are stored with a functional, or process, orientation. For example, data may be stored for invoices, payments, credit amounts, and so on.	Data are stored with a subject orientation that facilities, multiple views of the data and facilitates decision making .For example, sales may be recorded by product, by division, by manager, or by region.
Time-variant	Data are recorded as current transactions. For example, the sales data may be the sale of a product on a given date, such as Rs. 342.78 on 12-AUG-1999.	Data are recorded with a historical perspective in mind. Therefore, a time dimension is added to facilitate data analysis and various time comparisons.
Non-volatile	Data updates are frequent and common. For example, an inventory amount changes with each sale. Therefore, the data environment is fluid.	Data cannot be changed. Data are only added periodically from historical systems. Once the data are properly stored, no changes are allowed. Therefore the data environment is relatively static.

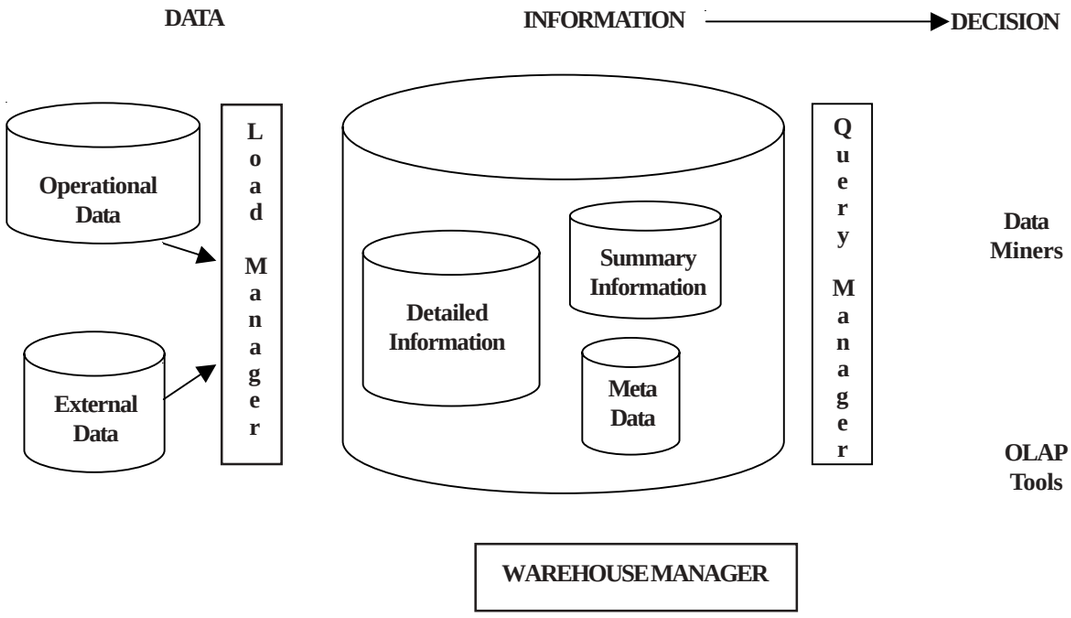


Fig. 17.2: Data Warehouse Architecture

The Load Manager

Data flows into the data warehouse through the *load manager*. The data is mostly extracted from the operational database(s) and other internal sources (like archived historical data), and supplemented by data imported from external sources. Externally sourced data can greatly enhance the value of information generated from a data warehouse. For example Transco, the gas pipeline operator in UK, uses weather forecast data from the British Met Office on a regular basis to determine demand for gas (the main source of energy used for heating homes and offices) in various areas of the country. The weather data is fed into a *model* that incorporates several other factors (e.g. day of the week, internal data about customers' usage patterns, demographic and economic profile data, alternate sources of energy, types of buildings in the area) to arrive at a demand forecast. Types of data from external sources that may be included in data warehouse are: financial indicators and statistics of the industry, market share data of competitors, demographic data, weather data, credit worthiness data, readership / viewer survey data for advertising media, specially commissioned surveys and so on. External data is usually obtained from commercial database services or government agencies (e.g. Equifax, Reuters, Met Office, census agency, industry associations, stock exchanges, local government statistics service). The data from such diverse sources will obviously be in different incompatible formats and will be distributed through various media. Some of them may be available on a downloadable format on the Internet; others may be distributed on CD-ROMs, while some may only be available on printed media. Some data may be available for free but most data (particularly when used for commercial purposes) have to be purchased.

The load manager primarily performs what is termed an *extract-transform-load (ETL)* operation.

- Data Extraction
- Data Transformation
- Data Loading

Data Extraction: This function has to deal with numerous data sources. Appropriate techniques have to be employed for each data source. Source data may be from different source machines in diverse data formats. Part of the source data may be in relational database systems. Some data may be on other legacy network and hierarchical data models. Many data sources may still be in flat files. There may also be the need to include data from spreadsheets and local departmental data sets. Data extraction can become quite a complex operation at times.

Various tools are available on the market for data extraction. Use of outside tools may be considered suitable for certain data sources. For the other data sources, in-house programs may need to be developed to do the data extraction. Purchasing outside tools may entail high initial costs. In-house programs, on the other hand, may mean ongoing costs for development and maintenance.

After extraction, the data needs to be kept somewhere for further preparation. Sometimes the extraction function is performed in the legacy platform itself if that approach suits the designed framework. More frequently, data warehouse implementation teams extract the source data into a separate physical environment from which moving the data into the data warehouse would be easier. In the separate environment, the source data may be extracted into a group of flat files, or an intermediate relational database, or a combination of both. This physical environment is called the *data-staging* area.

Data Transformation: In every system implementation, data conversion is an important function. For example, when implementing an operational system such as a magazine subscription application, the database has to be initially populated with data from the existing system records. The conversion may either be from a manual system or from a file-oriented system to a modern system supported with relational database tables. In either case, the data will need to be converted from the existing systems. So, what is so different for a data warehouse? Why is data transformation for a data warehouse more involved than that for an operational system?

As already discussed, data for a data warehouse comes from many disparate sources. If data extraction for a data warehouse poses great challenges, data transformation presents even greater challenges. Another factor in the data warehouse is that the data feed is not just an initial one-time load. The ongoing changes will have to continue to be picked up from the source systems. Any transformation tasks are set up for the initial load will have to be adapted for the ongoing revisions as well.

A number of individual tasks are performed as part of data transformation. First, the data extracted from each source is cleaned. Cleaning may be correction of misspellings, or may include resolutions of conflicts between state codes and pin codes in the source data, or may deal with providing default values for missing data elements, or elimination of duplicates when the same data is brought in from multiple source systems.

Standardization of data elements forms a large part of data transformation. The data types and field lengths for same data elements retrieved from the various sources need to be standardized. Semantic standardization is another major task. *Synonyms* and *homonyms* have to be resolved. Resolution of synonyms is required when two or more terms from different source systems mean the same thing. On the other hand, when a single term means many different things in different source systems, resolution of homonyms have to be performed.

Data transformation involves many forms of combining pieces of data from the different sources. In some cases, data from a single source record or related data elements from many source records are combined. In other situations, data transformation may also involve purging source data that is not useful and/or separating out source records into new combinations. During data transformation sorting and merging of data takes place on a large scale in the data staging area.

In many cases, the keys chosen for the operational systems are field values with built-in meanings. For example, the product key value may be a combination of characters indicating the product category, the code of the warehouse where the product is stored, and some code to show the production batch. Primary keys in the data warehouse cannot have built-in meanings. Therefore, data transformation also includes the assignment of surrogate keys derived from the source system primary keys.

A grocery chain point-of-sale operational system keeps the unit sales and revenue amounts by individual transactions at the checkout counter at each store. But in the data warehouse, it may not be necessary to keep the data at this detailed level. It may be more appropriate to summarize the totals by product at each store for a given day and keep the summary totals of the sale units and revenue in the data warehouse's storage. In such cases, the data transformation function would include such summarization processing.

The end result of the data transformation function is a collection of integrated data that is cleaned, standardized, and summarized. Now the data is ready to be loaded into each data set in the data warehouse.

Data Loading: Two distinct groups of tasks form the data loading function. After completion of the design and construction of the data warehouse, when it goes live for the first time, the initial loading of data is done. The initial load moves large volumes of data and takes substantial amount of time, but it is a one-time effort. As the data warehouse starts functioning, extraction of additions (and changes) to the source data continues on an ongoing basis, together with the transformation and loading operations.

The Query Manager

The *query manager* provides an interface between the data warehouse and its users. It performs tasks like directing the queries to the appropriate tables, generating views on an ad-hoc basis if required, monitoring the effectiveness of indexes and summary data, and query scheduling.

Data Warehouse Design Considerations

The key considerations involved in the design of a data warehouse are:

- Time Span
- Granularity
- Dimensionality
- Aggregations
- Partitioning

- **Time span:** Operational data represent current (atomic) transactions. Such transactions might define a purchase order, a sales invoice, an inventory movement, and so on. In short, operational data cover a short time frame. In contrast, data warehouse data tend to cover a longer time frame. Managers are seldom interested in a specific sales invoice to customer X; rather they tend to focus on sales generated during the last month, the last year, or the last five years. Rather than concern themselves with a single customer purchase, they might be interested in the buying pattern of such a customer or groups of customers. In short, data warehouse data tend to be historic in nature. That is, the data warehouse data represent company transactions up to a given point in time, yesterday, last week, last month, and the like. The time period for which data is held in the data warehouse is determined by the data analysis requirements of the users of the data warehouse. These needs, in turn, arise from the changes in the business environment that a particular organization needs to monitor, in its effort to stay ahead of its competitors. Since, a data warehouse's size depends on the span of time for which data is stored, the time span covered by the data warehouse is an important design consideration. If, for example, the environment changes rapidly, the data required for analysis would relate more often to the *recent past*, rather than that over several years or decades. In that case the designers of the data warehouse need to consider whether or not the cost incurred in holding data for indefinitely long time spans would be worthwhile.

- **Granularity:** According to Inmon, the single most important design aspect of a data warehouse is the decision on granularity. It refers to the level of detail or summarization available in units of data in the data warehouse. The more detail there is, the lower the level of granularity. The less detail there is, the higher the level of granularity.

Operational data represent specific transactions that occur at a given time, such as customer purchase of product X in store A. Thus, granularity is taken for granted to be of the lowest level, in operational systems. Data warehouse data must be presented at different levels of aggregation, from highly summarized to near atomic. This requirement is based on the fact that managers at different levels in the organization require data with different levels of aggregation. It is also possible that a single problem requires data with different summarization levels. For example, if a manager must analyze sales by region, (s)he must be able to access data showing the sales by region, by city within the region, by store within the city within the region, and so on. In this case, the manager requires summarized data to compare the regions, but (s)he also needs data in a structure that enables him or her to decompose (drill down) the data into more atomic components (that is, data at lower levels of aggregation). For example, it is necessary to be able to drill down to the stores within the region in order to compare store performance by region. Granularity level in a data warehouse cannot, therefore, be assumed.

The decision on granularity level profoundly affects both the volume of data that resides in the data warehouse, and the type of query that can be answered. A trade off exists between the volume of data in the data warehouse and the level of detail of queries (see *Figure 17.3* below). Some data warehouses are designed to support *dual granularity*. In such environments some data (usually the most recent) is held at a relatively low level of granularity, while the rest is held in more summarized form (i.e. at a higher granularity level). This enables detailed analysis at the same time allows reduction of data volume.

Fig. 17.3: Granularity of Data Warehouse Data

Fig. 17.4: Dimensionality of Data Warehouse Data

- **Dimensionality:** This is probably the most distinguishing characteristic of a data warehouse. From the data analyst's point of view, the data is always related in many different ways. For example, when we analyze product sales by a customer during a given time span, we are likely to ask how many widgets of type X were sold to customer Y during the last six months. In fact, the question tends to expand quickly to include many different data *dimensions*. For instance, we might want know how the product X fared relative to product Z during the past six months, by region, state, city, store, and customer (or sales of various products by quarters by country, as shown in *Figure 17.4*). In this case, both place and time are part of the picture. In general, data analysis tends to include many data dimensions, producing a multidimensional view of the data.

The data model used for modeling data warehouses is known as the *dimensional model*. The numerical measurements related to the business (like sales volumes) are stored in *fact tables*. The descriptions of the dimensions are stored in *dimension tables*. We will discuss this in more detail in later sections. The number and types of dimensions and facts that are to be stored in a data warehouse is a very important design decision, and (much like the decision on granularity) affects both the data volume and the types of analysis that can be supported.

- **Aggregations:** We have seen how data analysis queries directed at data warehouses involve dimensions. Another very common type of query directed at data warehouses involves sums of values along the different dimensions. For example: what is the total sales volume of VCR during the past 4 quarters? Answering this query using the fact and dimension tables would involve summing up the individual sales volume figures over the 4 quarters and the 3 counties. In real situations, similar queries might involve retrieving and summing hundreds or thousands of individual values. To avoid excessive processing load on the data warehouse arising from frequently asked queries of this type, it is often decided, at design time, to store some *pre-calculated aggregations*, along with the base facts, in the data warehouse (as illustrated in *Figure 17.4* above). This decision affects the data volume and performance of certain types of queries.
- **Partitioning:** One of the essences of the data warehouse is flexible data storage, management, and access. When data resides in large physical units, among other things it cannot be:
  - indexed easily
  - sequentially scanned, if needed
  - restructured easily
  - backed up conveniently
  - recovered easily
  - monitored easily

In short, having a big mass of data defeats much of the purpose of the data warehouse. The purpose of *partitioning* is to break the data into smaller (more manageable) physical units of storage. The criteria used for dividing the data can be: date, line of business / product category, geography / location, organizational / administrative unit, or any combination of these.

Activity A

What is the difference between a database and a data warehouse? Take some database from any organization and try to convert it into a data warehouse. What are the visible advantages that you can make out?

.....

.....

.....

17.5 DATA WAREHOUSE SCHEMA

One of the key questions to be answered by the database designer is: How can we design a database that allows unknown queries to be performant? This question encapsulates the differences between designing for a data warehouse and designing for an operational system. In a data warehouse one designs to support the *business process* rather than specific query requirements. In order to achieve this, the designer must understand the way in which the information within the data warehouse will be used.

In general, the queries directed at a data warehouse tend to ask questions about some essential fact, analyzed in different ways. For example reporting on:

- The average number of light bulbs sold per store over the past month
- The top ten most viewed cable-TV programs during the past week
- Top spending customers over the past quarter
- Customers with average credit card balances more than Rs.10,000 during the past year.

Each of these queries has one thing in common: they are all based on factual data. The content and presentation of the results may differ between examples, but the factual and transactional nature of the underlying data is the same.

Table 17.3: Fact Tables and Attributes

Requirement	Fact	Attributes
Sales of Light Bulbs	EPOS Transaction	Quanity Sold Product Identifier (SKU) Store Identifier Data and Time Revenue Achieved
Cable Programs	Cable Pay-per-view Transaction	Customer Identifier Cable Channel Watched Program Watched Data and Time Duration Household Identifier
Customer Spend	Loyalty Card Transaction	Customer Identifier Store Identifier Transaction Value Date and Time
Customer Account	Account Transactions	Customer Identifier Account Number Type of Transaction Destination Account Number

Fact data possesses some characteristics that allow the underlying information in the database to be structured. Facts are transactions that have occurred at some point in the past, and are unlikely to change in the future. Facts can be analyzed in different ways by cross-referencing the facts with different reference information.

For example, we can look at sales by store, sales by region, or sales by product. In a data warehouse facts also tend to have few attributes, because there are no operational data overheads. For each of the examples described, the attributes of the fact could be as listed in *Table 17.3* above.

One of the major technical challenges within the design of a data warehouse is to structure a solution that will be effective for a reasonable period of time (at least three to five years). This implies that the data should not have to be restructured when the business changes or the query profiles change. This is an important point, because in more traditional applications it is not uncommon to restructure the underlying data in order to address query performance issues.

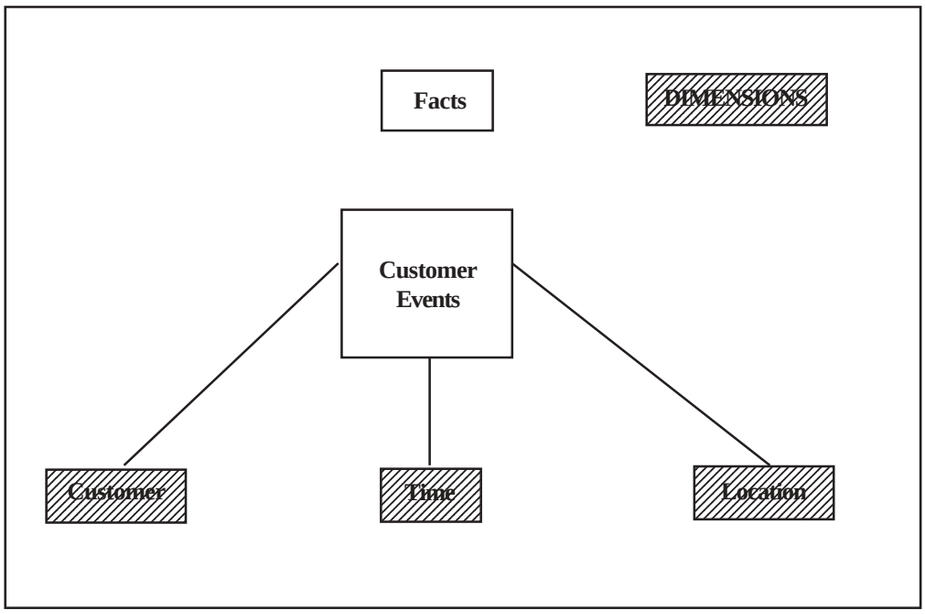


Fig. 17.5: Star Schema

The inherent advantage of factual transactions is that their content is unlikely to change, regardless of how it is analyzed. Also, the majority of the data volume of a data warehouse comes from the factual information. It is therefore possible to treat fact data as read-only data, and the reference data (used to interpret the fact data) as data that is liable to change over time. Thus, if reference data needs to change, the voluminous fact data would not have to be changed or restructured.

Star schemas (so called due to their ‘star like’ appearance) are physical database structures that store the factual data in the ‘center’, surrounded by the reference (or dimension) data (see *Figure 17.5* above).

The dimension tables should be relatively small (typically less than 5 GB in total) in comparison to the size of the data warehouse, so that restructuring costs are small as long as the keys to the fact tables are not changed. In star schema arrangements, the reference information is often denormalized to a single table to speed up query performance. The redundancy overheads are acceptable, as the sizes involved are small and even the reference information changes infrequently.

The dimensional information, represented by the reference data is often organized in form of *concept hierarchies* that are used for analysis, and designing data warehouse aggregations.

A typical concept hierarchy for a retail chain is depicted in *Figure 17.6* below.

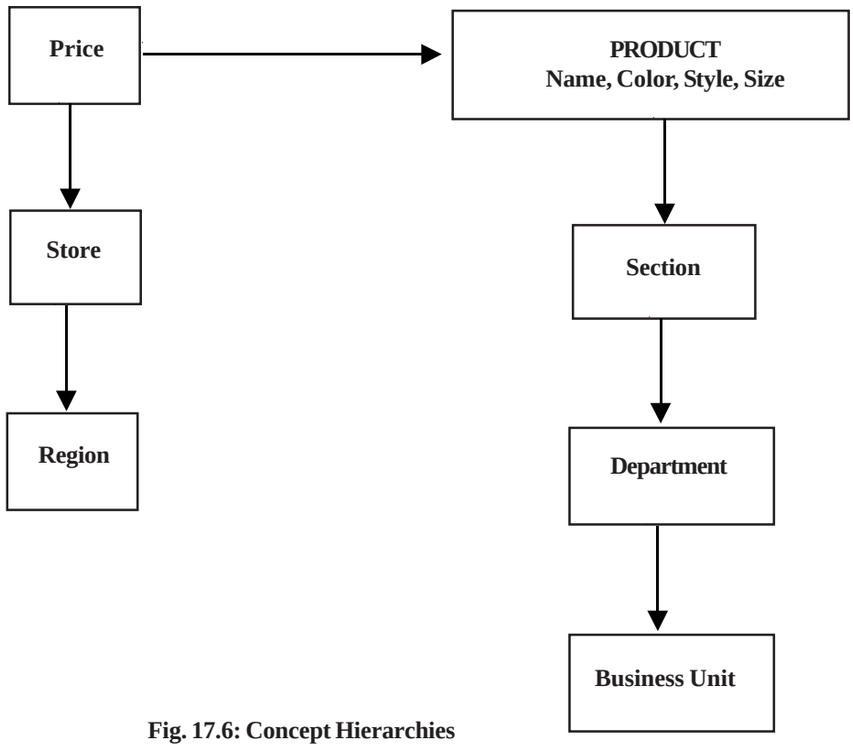


Fig. 17.6: Concept Hierarchies

The number of concept hierarchies that can be defined on any given dimension is by no means restricted to one. There may be several arbitrary concept hierarchies in use in any organization to enable data analysis from various angles. *Figure 17.7* below shows how the 'time' dimension in a retail data warehouse (represented by the day / date reference data) may be organized into several concept hierarchies or *groupings* (which are single level concept hierarchies – 'Easter' and 'Summer' in *Figure 17.7*)

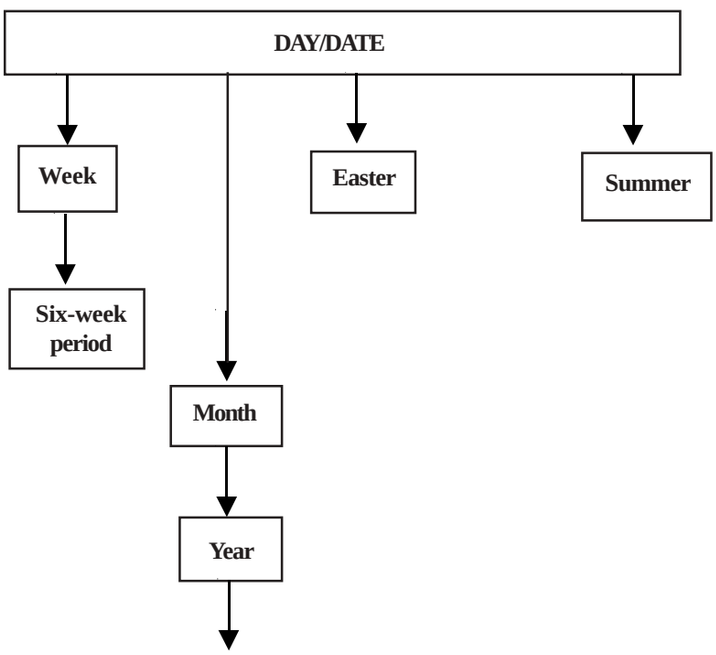
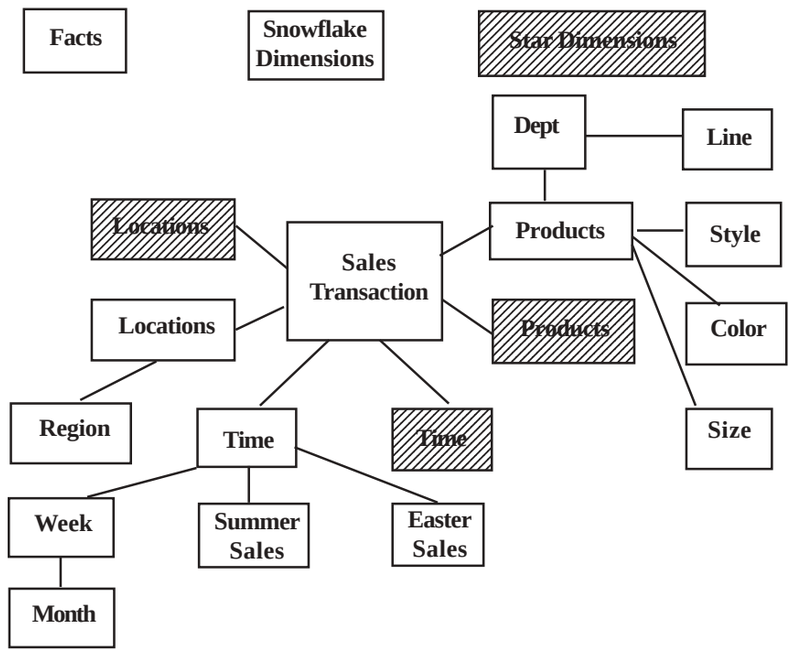


Fig. 17.7: Multiple Hierarchies

When the concept hierarchies and groupings are incorporated into a star schema diagram (like Figure 17.8 below), the appearance resembles a ‘snowflake’. Hence, schemas of this type are called *Snowflake schemas*.



Fi. 17.8: Snowflake Schema

17.6 METADATA

Metadata in a data warehouse is similar to the data dictionary in the context of a database. It stores data about data in the data warehouse.

Types of Metadata

Metadata in a data warehouse fall into three major categories:

- Operational Metadata
- Extraction and Transformation Metadata
- End-User Metadata

Operational Metadata: As already discussed, data for the data warehouse comes from several operational systems of the enterprise. These source systems contain different data structures. The data elements selected for the data warehouse have various field lengths and data types. Selecting data from different source files, and loading it into the data warehouse, requires splitting of records, combining parts of records from different source files, and dealing with multiple coding schemes and field lengths. When information is delivered to the end-users, it is essential to be able relate back to the original source data sets. Operational metadata contain all of this information about the operational data sources that allow us to trace back to the original source.

Extraction and Transformation Metadata: Extraction and transformation metadata contain data about the extraction of data from the source systems, namely, the extraction frequencies, extraction methods, and business rules for the data extraction. Also, this category of metadata contains information about all the data transformations that take place in the data staging area.

End-User Metadata. The end-user metadata is the navigational map of the data warehouse. It enables the end-users to find information from the data warehouse. The end-user metadata allows the end-users to use their own business terminology and look for information in those ways in which they normally think of the business.

Special Significance

Why is metadata especially important in a data warehouse?

- First, it acts as the glue that connects all parts of the data warehouse.
- Next, it provides information about the contents and structure to the developers.
- Finally, it opens the door to the end-users and makes the contents recognizable in their own terms.

Metadata Requirements

According to Inmon, a new user approaching a data warehouse wants to know

- What tables, attributes, and keys does the data warehouse contain?
- From where did each set of data come?
- What transformation logic was applied in loading the data?
- How has the data changed over time?
- What aliases exist, and how are they related to each other?
- What are the cross-references between technical and business terms? (For instance, the field name XVT-351J presumably meant something to a COBOL programmer in 1965, but what does it mean to me today?)
- How often does the data get reloaded?
- How much data is there? This helps end-users to avoid submitting unrealistic queries. Given some means of determining the size of tables, staff can tell the end users, “You can do what you like with 15,000 rows, but if it turns out be 15 million rows, back off and ask for help!”

Metadata requirements of various classes of users are summarized in *Table 17.4* below:

Table 17.4: Uses of Metadata

	IT Professionals	Power Users	Casual Users
Analysis and Discovery	Database Tables, Columns, Server Platforms.	Databases, Tables, Columns	List of Predefined Queries and Reports, Business views.
Meaning of Data	Data structures, Data Definitions, Data Mapping, Cleansing Functions, Transformation Rules	Business Terms, Data Definitions, Data Mapping, Cleansing Functions, Transformation Rules	Business Terms, Data Definitions, Filters, Data Sources, Conversion, Data Owners
Information Access	Program Code in SQL, 3GL, 4GL, Front-end Applications, Security	Query Toolsets, Database Access for Complex Analysis	Authorization Requests, Information Retrieval into Desktop Applications such as Spreadsheets.

Metadata Components

Warehouse metadata is not very different in kind from ordinary database metadata, although it is versioned in order to permit historical analysis. Prism gives the following breakdown of warehouse metadata in its Tech Topic, "Metadata in the Data Warehouse:"

Mapping

The mapping information records how data from operational sources is transformed on its way into the warehouse. Typical contents are:

- Identification of source fields
- Simple attribute-to-attribute mapping
- Attribute conversions
- Physical characteristic conversions
- Encoding/reference table conversions
- Naming changes
- Key changes
- Defaults
- Logic to choose from among multiple sources
- Algorithmic changes

Extract History

Whenever historical information is analyzed, meticulous update records have to be kept. The metadata history is a good place to start any time-based report, because the analyst has to know when the rules changed in order to apply the right rules to the right data. If, for example, sales territories were remapped in 1991, results from before that date may not be directly comparable with more recent results.

Miscellaneous

- Aliases can make the warehouses much more use-friendly by allowing a table to be queried by "Widgets produced by each factory" rather than "MF-STATS." Aliases also come in useful when different departments want to use their own names to refer to the same underlying data. Obviously, though, aliases can also cause a great deal of confusion if they are not carefully tracked.

- Often, parts of the same data warehouse may be in different stages of development. Status information can be used to keep track of this: for instance, tables might be classified "in-design," "in-test," inactive," or "active."

- Volumetric information lets users know how much data they are dealing with, so that they can have some idea how much their queries will cost in terms of time and computational resources. Volumetrics could usefully include such information as number of rows, growth rate, usage characteristics, indexing, and byte specifications.

- It is also useful to publish the criteria and time scales for purging old data.

Summarization and Aggregation Algorithms

As discussed above, a typical data warehouse contains lightly and heavily summarized data, and aggregations as well as full detailed records. The algorithms for summarizing (and aggregating) the detail data are obviously of interest to anyone who takes responsibility for interpreting the meaning of the summaries. This metadata can also save time by making it easier to decide which level of summarization is most appropriate for a given purpose.

Relationship Artifacts and History

Data warehouses implement relationships in a different way form production databases. Metadata pertaining to related tables, constraints, and cardinality are maintained, together with text descriptions and ownership records. This information and the history of changes to it can be useful to analysts.

Ownership / Stewardship

Operational databases are often owned by particular departments or business groups. In an enterprise data warehouse, however, all data is stored in a common format and accessible to all. This makes it necessary to identify the originator of each set of data, so that inquiries and corrections can be made by the proper group. It is useful to distinguish between *ownership* of data in the operational environment and *stewardship* in the data warehouse.

Access Patterns

It is desirable to record patterns of access to the warehouse in order to optimize and tune performance. Less frequently used data can be migrated to cheaper storage media, and various methods can be used to accelerate access to the data that is most in demand. Most databases do a good job of hiding such physical details, but specialized performance analysis tools are usually available. Some general-purpose tools, such as Information Builders' SiteAnalyzer, also are available.

Reference Tables / Encoded Data

Reference data is stored in an external table (see discussion on Star Schema above) and contains commonly used translations of encoded values. The contents of these tables must be stored in order to guarantee the ability to recover the original un-encoded data, together with effective from and effective to dates.

Data Model – Design Reference

Building a data warehouse without first constructing a data model is very difficult and frustrating. When a data model is used, metadata describing the mapping between the data model and the physical design should be stored. This allows all ambiguities or uncertainties to be resolved.

From the point of view of the Query Manager (see *Figure 17.2* above) of the data warehouse, the *Metadata Repository* can be perceived to have three logical layers: the *Information Navigator*, the *Business Metadata*, and the *Technical Metadata*.

Figure 17.9 below illustrates this concept. The query manager accesses the metadata through the Information Navigator layer which is the topmost layer of the metadata repository. The higher layers, in turn, access more detailed metadata components resident in the lower layers whenever required.

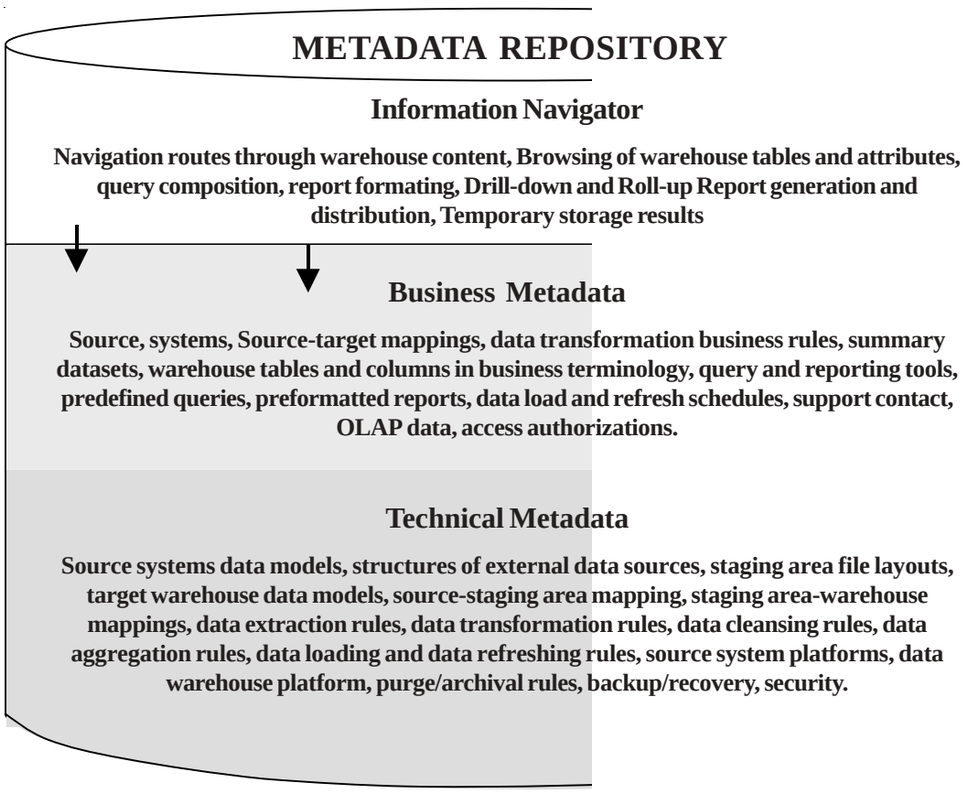


Fig. 17.9: The Metadata Repository

17.7 DATA WAREHOUSE AND THE WEB

Professor Peter Drucker, the senior guru of management practice, has admonished IT executives to look outside their enterprises for information. He remarked that the single biggest challenge is to organize outside data because change occurs from the outside. He predicted that the obsession with internal data would lead to being blindsided by external forces.

The majority of data warehousing efforts result in an enterprise focusing inward; however, the enterprise should be keenly alert to its externalities. As markets become turbulent, an enterprise must know more about its customers, suppliers, competitors, government agencies, and many other external factors. The changes that take place in the external environment, ultimately, get reflected in the internal data (and would be detected by the various data analysis tools discussed in the later sections), but by then it may be too late for the enterprise – proactive action is always better than reacting to external changes after the effects are felt. The conclusion is that the information from internal systems must be enhanced with external information. The synergism of the combination creates the greatest business benefits.

The importance of external data and the challenges faced in integrating external data

with internally sourced data was discussed in the Section on Load Manager above. In the same section, it was mentioned that, some externally sourced data (particularly time sensitive data), is often distributed through the internet.

Reliability of Web Content

Many question the reliability of web content, as they should. However, few analyze the reliability issue to any depth. The Web is a global bulletin board on which both the wise and foolish have equal space. Acquiring content from the Web should not reflect positively or negatively on its quality.

Consider the following situation: If you hear, “Buy IBM stock because it will double over the next month,” your reaction should depend on who made that statement and in what context. Was it a random conversation overheard on the subway, a chat with a friend over dinner, or a phone call from a trusted financial advisor? The *context* should also be considered when judging the reliability of Web content.

Think of Web resources in terms of quality and coverage, as shown in *Figure 17.10* below.

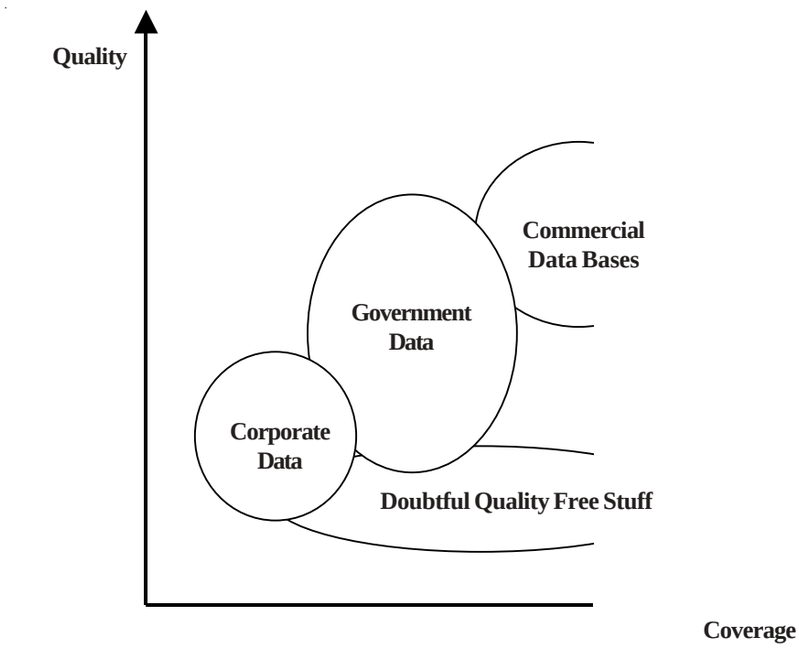


Fig. 17.10: Web-based Information Sources

Toward the top are information resources of high quality (accuracy, currency, and validity), and resources toward the right have a wide coverage (scope, variety, and diversity). The interesting aspect of the web is that information resources occupy all quadrants.

In the upper center, the commercial online database vendors traditionally have supplied business with high-quality information about numerous topics. However, the complexity of using these services and the infrequent update cycles have limited their usefulness.

More to the left, governmental databases have become tremendously useful in recent years. Public information was often available only by spending many hours of manual labour at libraries or government offices. Recent developments like the Electronic Data Gathering, Analysis, and Retrieval (EDGAR) database maintained by the U.S. Securities and Exchange Commission provide valuable and up-to-date data via the Web.

At the left, corporate Web sites often contain vast amounts of useful information in white papers, product demos, and press releases, eliminating the necessity to attend trade exhibits to learn the “latest and greatest” in a market place.

Finally, the “doubtful-quality free” content occupies the lower half of the figure. Its value is not in the quality of any specific item but in its constantly changing diversity. Combined with the other Web resources, the doubtful-quality free content acts as a wide-angle lens to avoid tunnel vision of the market place.

Web Farming

Like operational systems, the *Web farming* system provides input to the data warehouse. The result is to disseminate the refined information about specific business subjects to the enterprise sourced from the Web.

The primary source of content for the Web farming system is the Web because of its external perspectives on the business of the enterprise. As a content source, the Web can be supplemented (but not replaced) by the intranet web of the enterprise. This content is typically in the format of internal Web sites, word processing documents, spreadsheets, and e-mail messages. However, the content from the intranet is usually limited to internal information about the enterprise, thus negating an important aspect of Web farming.

Most information acquired by the Web farming system will not be in a form suitable for the data warehouse. Also, as discussed above, the source and quality of the content need to be judged. In any case, the information must be *refined* before loading into the warehouse. However, even in its unrefined state, the information obtained from the Web, through Web farming, could be highly valuable to the enterprise. The capability to directly disseminate this information may be required via textual message alerts or “What’s New” bulletins.

Refining Information

When a data warehouse is first implemented within an enterprise, a detailed analysis and reengineering of data from operational systems is required (see Section on Load Manager above). The same is true for Web farming. Before Web content can be loaded into a warehouse, the information must be refined.

The processes of refining information consists of four steps:

Discovery, *Acquisition*, *Structuring*, and *Dissemination*.

Discovery is the exploration of available Web resources to find those items that relate to specific topics. Discovery involves considerable detective work far beyond searching generic directory services, such as Yahoo!, or indexing services, such as Alta Vista. Further, the discovery activity must be a continuous process because data sources are continually appearing and disappearing from the Web.

Acquisition is the collection and maintenance of content identified by its source. The main goal of acquisition is to maintain the historical context so that you can analyze content in the context of its past. A mechanism to efficiently use human judgement in the validation of content is another key requirement.

Structuring is the analysis and transformation of content into a more useful format and into a more meaningful structure. The formats can be Web pages, spreadsheets, word processing documents, and database tables. As we move toward loading data into a warehouse, the structures must be compatible with the star-schema design and with key identifier values.

Dissemination is the packaging and delivery of information to the appropriate consumers, either directly or through a data warehouse. A range of dissemination mechanisms is required, from predetermined schedules to ad hoc queries. Newer technologies, such as information brokering and preference matching, may be desirable.

17.8 ON-LINE ANALYTICAL PROCESSING (OLAP)

In the previous sections we have discussed the need for building an Enterprise Data Warehouse, followed by discussions on its structure and ways and means of populating it with data. The data warehouse serves as the 'memory' of the enterprise. But, memory is of little use without intelligence. The tools that analyze data provide the 'intelligence'. In this section and the ones that follow, we discuss some of the tools available for accessing the data warehouse and transforming the data in it into information useful from the point of view of the business or *Business Intelligence*.

The term, On-line Analytical Processing (OLAP) was coined by E.F. Codd in 1993 to refer to a type of application that allows a user to interactively analyze data. Online Analytical Processing (OLAP) is a method of analyzing data in a multi-dimensional format, often across multiple time periods, with an aim of uncovering the business information concealed within the data. OLAP enables business users to gain an insight into the business through interactive analysis of different views of business data that have been built up from the operational systems. This approach facilitates a more intuitive and meaningful analysis of business information and assists in identifying important business trends.

OLAP can be defined as the process of converting raw data into business information through multi-dimensional analysis. This enables analysts to identify business strengths and weaknesses, business trends and the underlying causes of these trends. It provides an insight into the business through the interactive analysis of different views of business information, that have been built up from raw operating data which reflect the business users understanding of the business. The OLAP application contains logic, which includes multi-dimensional data selection, sub-setting of data, retrieval of data via the metadata layers and calculation formulas. The OLAP application layer is accessed via a front-end query tool, which uses tables and charts to "drill down" or navigate through dimensional data or aggregated measures (see *Figure 17.4* above).

Although OLAP applications are found in widely divergent functional areas, they all require the following key features: multidimensional views of data, calculation-intensive capabilities and time intelligence. *Figure 17.11* below illustrates the major features of an OLAP system.

Fig. 17.11: The OLAP System Architecture

The *OLAP Engine*, which is comprised of the '*Analytical processing logic*' and the '*Data processing logic*' layers, shown in Figure 11 above, provides a 'front-end' to the data warehouse – an interface through which a variety of general purpose *Graphical User Interface* (or *GUI*) type tools can access the data warehouse (see *Figure 17.12* below). This arrangement makes the data warehouse data available for non-multi-dimensional analysis as well.

Fig. 17.12: The OLAP Engine

The four major types of OLAP applications are multidimensional OLAP, hybrid OLAP, desktop OLAP and relational OLAP. Multidimensional OLAP is based on a multi-dimensional data base architecture. This stores data in a three-dimensional data cube that is already in the OLAP multidimensional format for "slicing and dicing" into analysis views. Relational OLAP products are designed to operate directly on a data warehouse built on relational databases, through a comprehensive metadata layer. Hybrid OLAP products primarily integrate specialized multidimensional data storage with relational database management technology. This allows businesses to link multi-dimensional data to the underlying source data in a relational database. The desktop style of OLAP allows users to perform limited analysis, directly against data held within a relational database, while avoiding many of the problems that affect hybrid and relational OLAP styles.

17.9 DATA VISUALIZATION

Business decision-makers, use data to identify opportunities, trends, and areas of concern in their respective businesses. Most data reaches the users in the form of tabular reports, which they find challenging to quickly and effectively absorb the information, spot patterns, identify aberrations, and see hidden relationships. On the other hand, the trained human eye can easily 'spot' patterns that the most advanced analytical tools would not be able to detect. Fortunately, though the volume of data that the users need to deal with is ever expanding, data-visualization tools have been evolving to the point that they can now transform large quantities of complex data into meaningful visual representations that incorporate the science behind human perception and cognition. As the old saying goes, a picture is often worth a thousand

words, or in this case, a thousand rows of data. Data-visualization applications render large quantities of data in the form of basic charts, graphical indicators, scorecards, dashboards, advanced visualizations, animations, and virtual reality environments. By automating the fundamental aspects of data analysis, they help information users identify trends and patterns in data that are often not apparent in traditional tabular representations. Managers have found these applications very effective for viewing business activity at a glance.

Data visualization involves graphical image generation by software, where the content of the image is determined by reading digital data. The data is usually numeric, but some software can visualize concepts drawn from text documents. The software arranges geometric shapes (such as points, lines, circles and rectangles) to create an interpretation of the data it read. Attributes such as proximity, size and color express relationships between the geometric shapes. For example, Figure 13 below visualized Product, Location, Revenue, Number of items sold, and Distribution channel information sourced from the data warehouse of a retail chain, in a single view. Data visualization has gained adoption among business users because it supports a variety of business tasks, including decision-making, knowledge management and business performance management.

Three strong trends have shaped the direction of data visualization software for business users over the last few years:

Chart Types. Most data visualizations depend on a standard chart type, whether a rudimentary pie chart or an advanced scatter plot (like the one used in Figure 17.13 below). The list of chart types supported by software has lengthened considerably in recent years.

Fig. 17.13: Visualization of Retail Data

Level of User Interaction with Visualization. A few years ago, most visualizations were static charts for viewing only. At the cutting edge of data visualization today, dynamic chart types are themselves the user interface, where the user manipulates a visualization directly and interactively to discover new views of information online.

Size and Complexity of Data Structures Represented by Visualization. A rudimentary pie or bar chart visualizes a simple series of numeric data points. Newer advanced chart types, however, visualize thousands of data points or complex data structures, such as neural nets.

Figure 17.14 depicts these trends and places in their context some of the common functionality of data visualization software. These trends also reveal a progression from forms of rudimentary data visualization (RDV) to advanced ones (ADV), as seen moving from lower-left to upper-right corners in Figure 1. Rudimentary forms of data visualization (pie and bar charts and presentation graphics) have been available in software for many years, whereas advanced forms (with interactive user interfaces, drill-down and live data connectivity) are relatively new. The dotted lines in Figure 14 create a life-cycle context for rudimentary and advanced forms of data visualization by identifying three life-cycle stages: maturing, evolving and emerging.

Fig. 17.14: Charting- State-of-the-art

The wide array of visualization software supports many different chart types, because the needs of users vary greatly. For example, business users demand pie and bar charts, whereas scientific users need scatter plots and constellation graphs. Users looking at geospatial data need maps and other three-dimensional data representations. Digital dashboards are popular with executive business intelligence users who monitor organizational performance metrics, visualizing them as speedometers, thermometers or traffic lights.

Tools for charting and presentation graphics are devoted exclusively to visualizing data. However, data visualization capabilities are commonly embedded in a wide range of software types, including tools for reporting, OLAP, text mining and data mining as well as applications for customer relationship management and business performance management.

Business software for charting has evolved from static and superficial charts to interactive visualizations with data connectivity and drill down. These advanced capabilities are found in a new category of software the enterprise charting system (ECS). With an ECS, users can develop and deploy chart-centric analytic applications that provide data visualization specifically for business intelligence on an enterprise scale.

Data visualization has long been in use in software for the study of mathematical, statistical, geographic and spatial data. Some data visualization software for business users has borrowed chart types originally designed for scientific users, such as scatter plots and constellation graphs.

Whereas scientific users tolerate heavily technical functionality that may require knowledge of programming languages or statistics, business users need this functionality to be hidden under a friendly user interface. For data visualization to appeal to business users, it must provide out-of-the-box value in the form of functionality for solving business problems, such as analyzing or mining customer behavior, product categories and business performance.

What is good visualization?

- **Effective:** ease of interpretation
- **Accurate:** sufficient for correct quantitative evaluation.
- **Aesthetics:** must not offend viewer's senses
- **Adaptable:** can adjust to serve multiple needs

Human perception capabilities and the fact that all displays need to be mapped to a 2-dimensional flat computer screen or sheets of paper (when printed) impose many limitations. For example, according to one researcher, inaccuracy of human perception increases in the following order when the respective techniques are used for visualization: Position along a common scale, Position along identical non-aligned scales, Length, Angle / slope, Area, Volume, Hue / saturation / intensity.

The techniques commonly used to overcome the limitations of 2-dimentional displays and size restrictions are:

- Scaling and offset to fit in range
- Derived values (e.g. residuals, logs) to emphasize changes
- Projections and other combinations to compress information
- Random jiggling to separate overlaps
- Multiple views to handle hidden relations, high dimensions
- Effective grids, keys and labels to aid understanding.

17.10 DATA MINING

According to Berry and Linoff, *Data Mining* is the exploration and analysis, by automatic or semiautomatic means, of large quantities of data in order to discover meaningful patterns and rules. This definition, justifiably, raises the question: how does data mining differ from OLAP? OLAP is undoubtedly a semiautomatic means of analyzing data, but the main difference lies in quantities of data that can be handled. There are other differences as well. Tables 5 and 6 summarize these differences.

Table 17.5: OLAP Vs Data Mining – Past Vs Future

OLAP: Report on the past	Data Mining: Predict the future
Who are our top 100 best customers for the last three years?	Which 100 customers offer the best profit potential?
Which customers defaulted on the mortgages last in two years?	Which customers are likely to be bad credit risks?
What were the sales by territory last quarter compared to the targets?	What are the anticipated sales by territory and region for next year?
Which salespersons sold more than their quota during last four quarters?	Which salespersons are expected to exceed their quotas next year?
Last year, which stores exceeded the total prior year sales?	For the next two years, which stores are likely to have best performance?
Last year, which were the top five promotions that performed well?	What is the expected return for next year's promotions?
Which customers switched to other phone companies last year?	Which customers are likely to switch to the competition next year?

Table 17.6: Differences between OLAP and Data Mining

FEATURES	OLAP	DATA MINING
MOTIVATION FOR INFORMATION REQUEST	What is happening in the enterprise?	Predict the future based only why this is happening
DATA GRANULARITY	Summary data	Detailed transaction-level data.
NUMBER OF BUSINESS DIMENSIONS	Limited number of dimensions	Large number of dimensions.
NUMBER OF DIMENSION ATTRIBUTES	Small number of attributes	Many dimension attributes
SIZES OF DATASETS FOR THE DIMENSIONS	Not large for each dimension	Usually very large for each dimension
ANALYSIS APPROACH	User-driven, interactive analysis	Data-driven automatic knowledge discovery
ANALYSIS TECHNIQUES	Multidimensional, drill-down, and slice-and-dice	Prepare data, launch mining tool and sit back
STATE OF THE TECHNOLOGY	Mature and widely used	Still emerging; some parts of the technology more mature

Why Now?

Why is data mining being put to use in more and more businesses? Here are some basic reasons:

- In today's world, an organization generates more information in a week than most people can read in a lifetime. It is humanly impossible to study, decipher, and interpret all that data to find useful patterns.
- A data warehouse pools all the data after proper transformation and cleansing into well-organized data structures. Nevertheless, the sheer volume of data makes it impossible for anyone to use analysis and query tools to discern useful patterns.

- In recent times, many data mining tools suitable for a wide range of applications have appeared in the market. The tools and products are now mature enough for business use.
- Data mining needs substantial computing power. Parallel hardware, databases, and other powerful components are available and are becoming very affordable.
- Organizations are placing enormous emphasis on building sound customer relationships, and for good reasons. Companies want to know how they can sell more to existing customers. Organizations are interested in determining which of their customers will prove to be of long-term value to them. Companies need to discover any existing natural classifications among their customers so that the each such class may be properly targeted with products and services. Data mining enables companies to find answers and discover patterns in their customer data.
- Finally, competitive considerations weigh heavily on organizations to get into data mining. Perhaps competitors are already using data mining.

Data Mining Techniques

Data mining covers a broad range of techniques. Each technique has been heavily researched in recent years, and several mature and efficient algorithms have evolved for each of them. The main techniques are: *Cluster detection*, *Decision trees*, *Memory based reasoning*, *Link analysis*, *Rule induction*, *Association rule discovery*, *Outlier detection and analysis*, *Neural networks*, *Genetic algorithms*, and *Sequential pattern discovery*. Discussion on the algorithms associated with the various techniques has been kept outside the scope of this text for two main reasons: firstly, because they are too mathematical / technical in nature, and secondly, because there are numerous, well written text books, to serve the needs of those who are specially interested in the subject. *Table 17.7* below summarized the important features of some of these techniques. The model structure refers to how the technique is perceived, not how it is actually implemented. For example, a decision tree model may actually be implemented through SQL statements. In the framework, the basic process is the process performed by the particular data mining technique. For example, the decision trees perform the process of splitting at decision points. How a technique validate the model is important. In the case of neural networks, the technique does not contain a validation method to determine termination. The model calls for processing the input records through the different layers of nodes and terminate the discovery at the output node.

Table 17.7: Summary of Data Mining Techniques

Data Mining Technique	Underlying Structure	Basic Process	Validation Method
Cluster Detection	Distance calculation in n-vector space	Grouping of values in the same neighborhood	Cross Validation to Verify Accuracy
Decision Trees	n-ary Tree	Splits at decision points based on entropy	Cross Validation
Memory-based Reasoning	Predictive Structure Based on Distance and Combination Functions	Association of unknown instances with known instances	Cross Validation
Link Analysis	Graphs	Discover links among variables by their values	Not Applicable
Neural Networks	Forward Propagation Network	Weighted inputs of predictors at each node	Not Applicable
Genetic Algorithms	Fitness Functions	Survival of the fittest on mutation of derived values	Mostly Cross Validation

Data Mining Applications

Data mining technology encompasses a rich collection of proven techniques that cover a wide range of applications in both the commercial and noncommercial realms. In some cases, multiple techniques are used, back to back, to greater advantage. For instance, a cluster detection technique to identify clusters of customers may be followed by a predictive algorithm applied to some of the identified clusters to discover the expected behaviour of the customers in those clusters.

Noncommercial use of data mining is strong and pervasive in the research area. In oil exploration and research, data mining techniques discover locations suitable for drilling based on potential mineral and oil deposits. Pattern discovery and matching techniques have military applications in assisting to identify targets. Medical research is a field ripe for data mining. The technology helps researchers with discoveries of correlations between diseases and patient characteristics. Crime investigation agencies use the technology to connect criminal profiles to crimes. In astronomy and cosmology, data mining helps predict cosmetic events.

The scientific community makes use of data mining to a moderate extent, but the technology has widespread applications in the commercial arena. Most of the tools target the commercial sector. Consider the following list of a few major applications of data mining in the business area.

Customer Segmentation: This is one of the most widespread applications. Businesses use data mining to understand their customers. Cluster detection algorithms discover clusters of customers sharing the same characteristics.

Market Basket Analysis: This very useful application for the retail industry. Association rule algorithms uncover affinities between products that are bought together. Other businesses such as upscale auction houses use these algorithms to find customers to whom they can sell higher-value items.

Risk Management: Insurance companies and mortgage businesses use data mining to uncover risks associated with potential customers.

Fraud Detection: Credit card companies use data mining to discover abnormal spending patterns of customers. Such patterns can expose fraudulent use of the cards.

Delinquency Tracking: Loan companies use the technology to track customers who are likely to default on repayments.

Demand Prediction: Retail and other businesses use data mining to match demand and supply trends to forecast for specific products.

Table 17.8: Application of Data Mining Techniques

Application Area	Examples of Mining Functions	Mining Processes	Mining Techniques
Fraud Detection	Credit Card Frauds Internal Audits Warehouse Pilferage	Determination of Variation from Norms	Data Visualization Memory-based Reasoning Outlier Detection and Analysis
Risk Management	Credit Card Upgrades Mortgage Loans Customer Retention Credit Rating	Detection and Analysis of Association Affinity Grouping	Decision Trees Memory Based Reasoning Neural Networks
Market Analysis	Market basket analysis Target marketing Cross selling Customer Relationship Management	Predictive Modeling Database Segmentation	Cluster Detection Decision Trees Association Rules Genetic Algorithms

The Data Mining Project

Step 1: Define Business Objectives— This step is similar to any information system project. First of all, determine whether a data mining solution is really needed. State the objectives. Are we looking to improve our direct marketing campaigns? Do we want to detect fraud in credit card usage? Are we looking for associations between products that sell together? In this step, define expectations. Express how the final results will be presented and used.

Step 2: Prepare Data— This step consists of data selection, preprocessing of data, and data transformation. Select the data to be extracted from the data warehouse. Use the business objectives to determine what data has to be selected. Include appropriate metadata about the selected data. Select the appropriate data mining technique(s) and algorithm(s). The mining algorithm has a bearing on data selection.

Unless the data is extracted from the data warehouse, when it is assumed that the data is already cleansed, pre-processing may be required to cleanse the data. Preprocessing could also involve enriching the selected data with external data. In the preprocessing sub-step, remove noisy data, that is, data blatantly out of range. Also ensure that there are no missing values.

Step 3: Perform Data Mining— Obviously, this is the crucial step. The knowledge discovery engine applies the selected algorithm to the prepared data. The output from this step is a set of relationships or patterns. However, this step and the next step of evaluation may be performed in an iterative manner. After an initial evaluation, there may be need to adjust the data and redo this step. The duration and intensity of this step depend on the type of data mining application. If the database is being segmented not too many iterations are needed. If a predictive model is being created, the models are repeatedly set up and tested with sample data before testing with the real database.

Step 4: Evaluate Results— The aim is to discover interesting patterns or relationships that help in the understanding of customers, products, profits, and markets. In the selected data, there are potentially many patterns or relationships. In this step, all the resulting patterns are examined, and a filtering mechanism is applied so as to select only the promising patterns for presentation and use.

Step 5: Present Discoveries— Presentation of patterns / associations discovered may be in the form of visual navigation, charts, graphs, or free-form texts. Presentation also includes storing of interesting discoveries in the knowledge base for repeated use.

Step 6: Ensure Usage of Discoveries— The goal of any data mining operation is to understand the business, discern new patterns and possibilities, and also turn this understanding into actions. This step is for using the results to create actionable items in the business. The results of the discovery are disseminated so that action can be taken to improve the business.

Selecting Data Mining Software Tools

Before we get into a detailed list of criteria for selecting data mining tools, let us make a few general but important observations about tool selection.

- The tool must be able to integrate well with the data warehouse environment by accepting data from the warehouse and be compatible with the overall metadata framework.
- The patterns and relationships discovered must be as accurate as possible. Discovering erratic patterns is more dangerous than not discovering any patterns at all.

- In most cases, an explanation for the working of the model and how the results were produced is required. The tool must be able to explain the rules and how patterns were discovered.

Let us complete this section with a list of criteria for evaluating data mining tools. The list is by no means exhaustive, but it covers the essential points.

Data Access: The data mining tool must be able to access data sources including the data warehouse and quickly bring over the required datasets to its environment. On many occasions data from other sources may be needed to augment the data extracted from the data warehouse. The tool must be capable of reading other data sources and input formats.

Data Selection: While selecting and extracting data for mining, the tool must be able to perform its operations according to a variety of criteria. Selection abilities must include filtering out of unwanted data and deriving new data items from existing ones.

Sensitivity to Data Quality: Because of its importance, data quality is worth mentioning again. The tool must be able to recognize missing or incomplete data and compensate for the problem. The tool must also be able to produce error reports.

Data Visualization: Data mining techniques process substantial data volumes and produce a wide range of results. Inability to display results graphically and diagrammatically diminishes the value of the tool severely.

Extensibility: The tool architecture must be able to integrate with the data warehouse administration and other functions such as data extraction and metadata management.

Performance: The tool must provide consistent performance irrespective of the amount of data to be mined, the specific algorithm applied, the number of variables specified, and the level of accuracy demanded.

Scalability: Data mining needs to work with large volumes of data to discover meaningful and useful patterns and relationships. Therefore, ensure that the tool scales up to handle huge data volumes.

Openness: This is a desirable feature. Openness refers to being able to integrate with the environment and other types of tools. The ability of the tool to connect to external applications where users could gain access to data mining algorithms from the applications, is desirable. The tool must be able to share the output with desktop tools such as graphical displays, spreadsheets, and database utilities. The feature of openness must also include availability of the tool on leading server platforms.

Suite of Algorithms: A tool that provides several different algorithms rather than one that supports only a few data mining algorithms, is more advantageous.

Multi-technique Support: A data mining tool supporting more than one technique is worth consideration. The organization may not presently need a composite tool with many techniques, but a multi-technique tool opens up more possibilities. Moreover, many data mining analysts desire to cross-validate discovered patterns using several techniques.

Activity B

You are a data mining consultant in a large commercial bank that provides many financial services. The bank already has a data warehouse which it rolled out two years ago. The management wants to find the existing customers who are likely to respond to a marketing campaign offering new services.

a) Outline the data mining process, list the steps / phases, and indicate the activities in each phase.

b) In the project you are responsible for analyzing the requirements and selecting a toolset / software for data mining. Make a list of the criteria you will use for the toolset / software selection. Briefly explain why each criterion is necessary.

c) What could be the business benefits of the exercise? Would the effort and cost be justified?

.....

.....

.....

.....

.....

.....

.....

.....

Benefits of Data Mining

Without data mining, useful knowledge lying buried in the mountains of data in many organizations would never be discovered and the benefits from using the discovered patterns and relationships would not be realized. What are the types of such benefits? We have already touched upon the applications of data mining and have indicated the implied benefits.

Just to appreciate the enormous utility of data mining, the following list enumerates some real-world situations:

- In a large company manufacturing consumer goods, the shipping department regularly short-ships orders and hides the variations between the purchase orders and the freight bills. Data mining detects the criminal behaviour by uncovering patterns of orders and premature inventory reductions.
- A mail order company improves direct mail promotions to prospects through more targeted campaigns.
- A supermarket chain improves earnings by rearranging the shelves based on discovery of affinities of products that sell together.
- An airlines company increases sales to business travelers by discovering traveling patterns of frequent flyers.
- A department store hikes the sales in specialty departments by anticipating sudden surges in demand.
- A national health insurance provider saves large amounts of money by detecting fraudulent claims.
- A major banking corporation with investment and financial services increases the leverage of direct marketing campaigns. Predictive modeling algorithms uncover clusters of customers with high lifetime values.
- A manufacturer of diesel engines increases sales by forecasting sales of engines based on patterns discovered from historical data of truck registrations.
- A major bank prevents loss by detecting early warning signs for attrition in its checking account business.
- A catalog sales company doubles its holiday sales from the previous year by predicting which customers would use the holiday catalog.

17.11 SUMMARY

Information for strategic decision-making is of utmost importance to organizations. Due to several practical difficulties with the structure and nature of data held in operational systems in generating the required information, a new concept known as data warehousing has emerged. In the data warehousing environment data collected from internal operational systems, as well as data from external sources is loaded into a subject-oriented, time-variant, integrated, and non-volatile environment. The data warehouse environment is carefully designed with time-span, granularity, aggregations, dimensionality and partitioning in mind. The initial data feed to the data warehouse as well as regular loading of data takes place in three phases: extraction, transformation and loading, and is done by Load Manager component of the data warehouse. The data in the data warehouse is organized in terms of fact data and dimension data. The star and snowflake schemas used in data warehouses support the separation of fact data and dimension data, as well as efficient query processing. Metadata or data about data is an indispensable and integral component of a data warehouse. Data collected from external sources can supplement internal data and prove extremely valuable. The concept of collecting information from the Web primarily for loading into an organization's data warehouse is known as web farming. On-line analytical processing (OLAP) technology provides multi-dimensional analysis capability to the users of a data warehouse. It also provides an interface through which other client software can access the data warehouse. Data visualization technology has made great progress in recent years. It can act as an important tool in the hands of trained users, by providing valuable support to spotting trends and / or deviations in data. Apart from OLAP and data visualization tools that are user-driven semiautomatic tools for data analysis, another class of tools, that are more automatic in their analysis and at the same time capable of accessing and using large volumes of data, have matured recently. This category of tools, known as data mining tools, have found applicability in many business area. A variety of data mining techniques and algorithms are available for data exploration and discovery of 'novel' patterns and relationships. The business benefits derived from the discoveries made by the data mining tools could be immense.

17.12 UNIT END EXERCISES

1) What are the practical difficulties of using an operational system's database to service the information needs of strategic decision-making?

2) Briefly explain the importance of the Extract-Transform-Load operation, and the functions of the Load Manager component of a data warehouse.

3) Explain briefly the terms:
   a) subject orientation
   b) granularity
   c) aggregations
   d) partitioning
   e) time variancy.

4) In the context of a data warehouse:
   a) what is a concept hierarchy?
   b) what do you mean by factual information?
   c) what is a snowflake schema?

5) Why is it important to separate fact data from reference data in a data warehouse?

6) Why is metadata an important component of a data warehouse? What is a metadata repository and how is it used by the query manager component of a data warehouse?

7) What do you mean by 'business intelligence'? What is an OLAP Engine?

8) Explain how the progress of data visualization technology in recent years has helped data analysts and business decision makers.

9) How can you use the Web as a data source for your data warehouse? What types of information can you get from the Web? Explain briefly the steps needed to ensure that only good quality, reliable data is loaded into the data warehouse from the Web.

10) How is data mining different from OLAP? Explain briefly.

17.13 REFERENCES AND SUGGESTED FURTHER READINGS

Adriaans, Pieter, Zantinge, Dolf, 2000, *Data Mining*, Addison Wesley Longman Ltd., Harlow, England

Berry, Michael J.A., Linoff, Gordon, 2001, *Mastering Data Mining*, John Wiley & Sons, Inc., New York

Inmon, W.H., 1994, *Using the Data Warehouse*, John Wiley & Sons (Asia) Pte, Ltd., Singapore

Inmon, W.H., 1996, *Building the Data Warehouse*, John Wiley & Sons (Asia) Pte, Ltd., Singapore

Inmon, W.H., Welch, J.D., Glassey, K.L., 1996, *Managing the Data Warehouse*, John Wiley & Sons, Inc., New York

Leon, Alexis, Leon, Mathews, 1999, *Database Management Systems*, Leon Vikas, Chennai

Sinha, Amitesh, 2002. *Data Warehousing*, Thomson Asia Pte. Ltd., Singapore

Ponniah Paulraj, 2003, *Data Warehousing Fundamentals*, John Wiley & Sons (Asia) Pte, Ltd., Singapore

Turban E, Mclean E, Wetherbe J, 2004, *Information Technology for Management, 4th Ed,.* John Wiley & Sons (Asia) Pte, Ltd., Singapore